UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/492,544	01/27/2000	Michael K. Gschwind	Y0999-357(8728-320)	1007

46069 7590 08/09/2005

F. CHAU & ASSOCIATES, LLC
130 WOODBURY ROAD
WOODBURY, NY 11797

EXAMINER	
MEONSKE, TONIA L	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 08/09/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/492,544	GSCHWIND, MICHAEL K.	
	Examiner	Art Unit	
	Tonia L. Meonske	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 May 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-31 and 33-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-31 and 33-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1, 3, 4 5, 10, 11, 31, and 33-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Katzman, US Patent 3,737,871, in view of Morris, US Patent 6,286,095 and Hamacher et al., Computer Organization, 1978, McGraw-Hill, Inc., second edition, pages 112-114 (herein referred to as Hamacher).

3. Referring to claims 1 and 3, Katzman has taught a method for renaming memory references to stack locations in a computer processing system, comprising the steps of:

a. detecting stack references that use architecturally defined stack access methods (Katzman, Abstract, column 1, lines 20-30, column 4, lines 36-67); and

b. replacing the stack references with references to registers (Katzman, column 4, lines 36-67, Column 4, line 35, column 6, line 22,).

4. Katzman has not specifically taught that the registers are processor-internal. However, Katzman has recognized that accessing the main memory is time consuming and that there is a need to speed up the access to the most frequently used data. Katzman has used the TOS registers, which are not in the main memory, to speed up the data access of the most frequently used data (column 1, lines 20-30, column 2, line 64-column 3, line 5, column 3, lines 24-39,

column 6, lines 25-30). Katzman has not specifically taught where the registers reside. However, Hamacher has taught that having the TOS registers stored inside the processor shortens the time for most accesses to the stack (pages 12-14). Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to have the registers of Katzman be processor-internal, as taught by Hamacher, for the desirable purpose of shortening the time for most accesses to the stack (pages 12-14).

5. Katzman has not specifically taught further comprising the step of synchronizing an architected state between the processor-internal registers and a main memory of the computer processing system wherein said synchronizing step comprises the step of inserting in-order write operations for all of the stack references that are write stack references. However, Morris et al. have taught synchronizing an architected state between the processor-internal registers and a main memory of the computer processing system wherein said synchronizing step comprises the step of inserting in-order write operations for all of the stack references that are write stack references (Morris et al., Abstract, Figures 4A, 4B, 11, and12, column 4, lines 9-37, column 5, lines 14-49) for the purpose of only operating on valid data (Morris et al., column 6, lines 62-67). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention of Katzman, include the step of synchronizing an architected state between the processor-internal registers and a main memory of the computer processing system wherein said synchronizing step comprises the step of inserting in-order write operations for all of the stack references that are write stack references, as taught by Morris et al., for the desirable purpose of only operating on valid data (Morris et al., column 6, lines 62-67).

6. Referring to claim 4, Katzman has taught the method according to claim 1, as described above, and further comprising the step of performing a consistency preserving operation for a stack reference that does not use the architecturally defined stack access methods (Katzman, column 4, line 35, column 4, lines 22-47).

7. Referring to claim 5, Katzman has taught the method according to claim 4, wherein said step of performing a consistency-preserving operation comprises the step of bypassing a value from a given processor-internal register to a load operation that references a stack area and that does not use the architecturally defined stack access methods (Katzman, column 4, line 35, column 4, lines 22-47).

8. Referring to claim 10, Katzman has taught the method according to claim 1, as described above, and wherein the architecturally defined stack access methods comprise memory accesses that use at least one of a stack pointer, a frame pointer, and an argument pointer (Katzman, column 3, 24-39, column 4, lines 42-48,).

9. Referring to claim 11, Katzman has taught the method according to claim 1, as described above, and wherein the architecturally defined stack access methods comprise push, pop, and other stack manipulation operations (Katzman, column 4, lines 33-36, column 7, lines 30-41).

10. Claim 31 does not recite limitations above the claimed invention set forth in claim 1 and is therefore rejected for the same reasons set forth in the rejection of claim 1 above.

11. Claim 33 does not recite limitations above the claimed invention set forth in claim 3 and is therefore rejected for the same reasons set forth in the rejection of claim 3 above.

12. Claim 34 does not recite limitations above the claimed invention set forth in claim 4 and is therefore rejected for the same reasons set forth in the rejection of claim 4 above.

13. Claim 35 does not recite limitations above the claimed invention set forth in claim 1 and is therefore rejected for the same reasons set forth in the rejection of claim 1 above.

14. Claim 36 does not recite limitations above the claimed invention set forth in claim 3 and is therefore rejected for the same reasons set forth in the rejection of claim 3 above.

15. Claims 6-9, and 37-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Katzman, US Patent 3,737,871, in view of Hamacher et al., Computer Organization, 1978, McGraw-Hill, Inc., second edition, pages 112-114 (herein referred to as Hamacher), Wing, US Patent 5,926,832 and Morris, US Patent 6,286,095.

16. Referring to claim 6, Katzman has taught the method according to claim 4, as described above. Katzman has not specifically taught and further comprising the step of synchronizing an architected state between the processor-internal registers and a main memory of the computer processing system, and wherein said step of performing a consistency-preserving operation comprises the step of recovering an in-order value for the stack reference from the main memory, upon performing said synchronizing step. However, Wing et al. have taught further comprising the step of synchronizing an architected state between the processor-internal registers and a main memory of the computer processing system (Wing et al., column 26, lines 5-52), and wherein said step of performing a consistency-preserving operation comprises the step of recovering an in-order value for the stack reference from the main memory, upon performing said synchronizing step (Wing et al., column 26, lines 5-52). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the system of Katzman include the step of synchronizing an architected state between the processor-internal registers and a main memory of the computer processing system, and wherein said step of performing a

consistency-preserving operation comprises the step of recovering an in-order value for the stack reference from the main memory, upon performing said synchronizing step, as taught by Wing et al., for the desirable purpose of maintaining consistent data in both memories, so that valid unstale data is used during execution (Wing et al., column 26, lines 5-52).

17. Referring to claim 7, Katzman, in combination with Wing et al. have taught the method according claim 6, as described above. They have not taught wherein the in-order value is written to the main memory by an in-order write operation inserted into an instruction stream containing an instruction corresponding to the stack reference, when the stack reference is a write stack reference. However, Morris et al. have taught the in-order value is written to the main memory by an in-order write operation inserted into an instruction stream containing an instruction corresponding to the stack reference, when the stack reference is a write stack reference (Morris et al., Abstract, Figures 4A, 4B, 11, and12, column 4, lines 9-37, column 5, lines 14-49) for the purpose of only operating on valid data (Morris et al., column 6, lines 62-67). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention of Katzman, include wherein the in-order value is written to the main memory by an in-order write operation inserted into an instruction stream containing an instruction corresponding to the stack reference, when the stack reference is a write stack reference, as taught by Morris et al. for the desirable purpose of only operating on valid data (Morris et al., column 6, lines 62-67).

18. Referring to claim 8, Katzman, in combination with Wing et al. have taught the method according to claim 6, further comprising the step of writing the in-order value to the main

memory in response to a load operation that does not use the architecturally defined stack access methods (Wing et al., column 26, lines 5-52).

19. Referring to claim 9, Katzman has taught the method according to claim 4, as described above. Katzman has not taught and wherein said step of performing a consistency-preserving operation comprises the steps of:

a. discarding all out-of-order state of a processor the system;

b. synchronizing an architected state between the processor-internal registers and a main memory of the computer processing system; and

c. restarting execution after a store operation has been performed that does not use the architecturally defined stack access methods.

20. However, Wing et al have taught:

a. discarding all out-of-order state of a processor the system (Wing et al., column 26, lines 24-30);

b. synchronizing an architected state between the processor-internal registers and a main memory of the computer processing system (Wing et al., column 26, lines 5-52); and

d. restarting execution after a store operation has been performed that does not use the architecturally defined stack access methods (Wing et al., column 26, lines 24-30).

21. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the system of Katzman, include the steps of discarding all out-of-order state of a processor the system; synchronizing an architected state between the processor-internal registers and a main memory of the computer processing system; and restarting execution after a

store operation has been performed that does not use the architecturally defined stack access methods, as taught by Wing et al., for the desirable purpose of recovering from memory inconsistencies which occur between memory and the execution, or rename, registers (Wing et al., column 26, lines 5-52).

22. Claim 37 does not recite limitations above the claimed invention set forth in claim 7 and is therefore rejected for the same reasons set forth in the rejection of claim 7 above.

23. Claim 38 does not recite limitations above the claimed invention set forth in claim 8 and is therefore rejected for the same reasons set forth in the rejection of claim 8 above.

24. Claim 39 does not recite limitations above the claimed invention set forth in claim 9 and is therefore rejected for the same reasons set forth in the rejection of claim 9 above.

25. Claims 12-19 and 21-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Evoy et al., US Patent 5953741 and Katzman, US Patent 3,737,871.

26. Referring to claim 12, Evoy et al. in combination with Katzman have taught a method for renaming memory references to stack locations in a computer processing system, comprising the steps of

a. determining whether a load instruction references a location in a local stack using an architecturally defined register for accessing a stack location (Katzman, column 4, lines 5-25, Column 4, line 35, column 6, line 22, column 4, line 35, column 6, lines 22-47);

b. determining whether a rename register exists for the referenced location in the local stack, when the load instruction references the location using the architecturally

defined register (Katzman, Abstract, column 1, lines 20-30, column 4, lines 36-67, column 4, line 35, column 4, lines 22-47); and

c. replacing the reference to the location by a reference to the rename register, when the rename register exists (Katzman, column 4, lines 36-67, column 4, line 35, column 4, lines 22-47).

27. Katzman has not specifically taught that this method for renaming memory references to stack locations is in a multiprocessor computer processing system wherein each processor comprises a respective local stack. However, Evoy et al. have taught a method for renaming memory references to stack locations in a multiprocessor computer processing system wherein each processor comprises a local stack. (column 4, lines 53-60, column 5, lines 2-27, column 5, line 60-column 6, line 14, column 7, lines 26-33, Each stack based processor contains a local stack, see Figure 2, elements 78 and 76.). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention of Evoy et al., include the steps of Katzman, as described above, in order to implement the TOS registers of Katzman, in each stack based processor of Evoy et al. for the desirable purpose of allowing the stack based processors of Evoy et al. to be operated at a maximum speed (Katzman, column 1, lines 20-30).

28. Referring to claim 13, Katzman and Evoy et al. have taught the method according to claim 12, as described above, and wherein the architecturally defined register corresponds to a pointer for accessing stack locations (Katzman, Abstract, column 1, lines 20-30, column 4, lines 36-67).

29. Referring to claim 14, Katzman and Evoy et al. have taught the method according to claim 13, as described above, and wherein the pointer for accessing the stack locations is one of

a stack pointer, a frame pointer, and an argument pointer (Katzman, column 3, lines 24-39, column 4, lines 42-48).

30. Referring to claim 15, Katzman and Evoy et al. have taught the method according to claim 12, as described above, and wherein the architecturally defined stack access methods comprise push, pop, and other stack manipulation operations (Katzman, column 4, lines 33-36, column 7, lines 30-41).

31. Referring to claim 16, Katzman and Evoy et al. have taught the method according to claim 12, as described above, and wherein said step of determining whether the renaming register exists comprises the step of computing one of a symbolic address and an actual address of the location (Katzman, Column 3, lines 55-67, Column 4, line 35, column 6, line 22).

32. Referring to claim 17, Katzman and Evoy et al. have taught the method according to claim 12, as described above, and wherein said step of determining whether the rename register exists is performed during one of a decode, an address generation, and a memory access phase (Katzman, Column 3, lines 55-67, Column 4, line 35, column 6, line 22).

33. Referring to claim 18, Katzman and Evoy et al. have taught the method according to claim 12, as described above, and further comprising the step of performing the load instruction from one of a main memory and a cache of the system, when the rename register does not exist (Katzman, Column 3, lines 55-67, Column 4, line 35, column 6, line 22, column 4, line 35, column 4, lines 22-47).

34. Referring to claim 19, Katzman and Evoy et al. have taught the method according to claim 12, as described above, and further comprising the step of determining whether the load instruction references a location in any stack, including the local stack, using another register,

when the load instruction does not reference the location using the architecturally defined register (Katzman, column 4, line 35, column 4, lines 22-47).

35. Referring to claim 21, Katzman and Evoy et al. have taught the method according to claim 19, as described above, and further comprising the step of performing the load instruction from one of a main memory and a cache of the system, when the load instruction does not reference the location using the other register (Katzman, column 6, lines 47-67).

36. Referring to claim 22, Katzman and Evoy et al. have taught the method according to claim 19, as described above, and further comprising the step of executing a consistency-preserving mechanism to perform the load instruction from the stack area, when the load instruction references the location using the other register (Katzman, column 4, line 35, column 4, lines 22-47).

37. Referring to claim 23, Katzman and Evoy et al. have taught the method according to claim 12, as described above, and further comprising the step of allocating a rename register for the location, when the rename register does not exist (Katzman, column 4, lines 36-67, Column 4, line 35, column 6, line 22).

38. Referring to claim 24, Katzman and Evoy et al. have taught the method according to claim 23, as described above, and further comprising the step of inserting an operation, into an instruction stream containing the load instruction, to load the location from a processor of the system to the allocated rename register, upon allocating the rename register (Katzman, column 7, lines 1-29, The operations inserted into the system to perform register renaming.).

39. Referring to claim 25, Katzman and Evoy et al. have taught the method according to claim 24, as described above, and further comprising the step of:

a. replacing the reference to the location by a reference to the allocated rename register, upon inserting the operation (Katzman, column 7, lines 1-29).

40. Referring to claim 26, Katzman has taught a method for renaming memory references to stack locations in a computer processing system, comprising the steps of:

a. determining whether a store instruction references a location in a local stack using an architecturally defined register for accessing a stack location (Katzman, column 4, lines 1-25, Abstract, column 1, lines 20-30, column 4, lines 36-67, Column 4, line 35, column 6, line 22, column 4, line 35, column 4, lines 22-47);

b. allocating a rename register for the location, when the store instruction references the location using the architecturally defined register (Katzman, column 4, lines 36-67, Column 4, line 35, column 6, line 22, column 4, line 35, column 4, lines 22-47); and

c. replacing the reference to the location by a reference to the rename register (Katzman, column 4, lines 36-67, Column 4, line 35, column 6, line 22, column 4, line 35, column 4, lines 22-47).

41. Katzman has not specifically taught that this method for renaming memory references to stack locations is in a multiprocessor computer processing system wherein each processor comprises a respective local stack. However, Evoy et al. have taught a method for renaming memory references to stack locations in a multiprocessor computer processing system wherein each processor comprises a local stack. (column 4, lines 53-60, column 5, lines 2-27column 5, line 60-column 6, line 14, column 7, lines 26-33, Each stack based processor contains a local stack, see Figure 2, elements 78 and 76.). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention of Evoy et al., include the steps

of Katzman, as described above, in order to implement the TOS registers of Katzman, in each stack based processor of Evoy et al. for the desirable purpose of allowing the stack based processors of Evoy et al. to be operated at a maximum speed (Katzman, column 1, lines 20-30).

42. Referring to claim 27, Katzman and Evoy et al. have taught the method according to claim 26, further comprising the step of inserting an operation, into an instruction stream containing the store instruction, to store the location from the rename register to a main memory of the system, upon replacing the reference to the location by the reference to the rename register (Katzman, column 7, lines 1-29, Column 4, line 35, column 6, line 22, The operations inserted into the system to perform register renaming.).

43. Claim 28 does not recite limitations above the claimed invention set forth in claim 19 and is therefore rejected for the same reasons set forth in the rejection of claim 19 above.

44. Claim 29 does not recite limitations above the claimed invention set forth in claim 21 and is therefore rejected for the same reasons set forth in the rejection of claim 21 above.

45. Claim 30 does not recite limitations above the claimed invention set forth in claim 22 and is therefore rejected for the same reasons set forth in the rejection of claim 22 above.

46. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Evoy et al. in view of Katzman, US Patent 3,737,871, and Wing et al., US Patent 5,926,832.

47. Referring to claim 20, Katzman has taught the method according to claim 19. Katzman has not taught wherein said step of determining whether the load instruction references the location using the other register comprises the step of marking translation lookaside buffer (TLB) entries of pages in the local stack as containing stack references, when the load instruction references the location using the other register. However, Wing et al. have taught wherein said

step of determining whether the load instruction references the location using the other register comprises the step of marking translation lookaside buffer (TLB) entries of pages in the local stack as containing stack references, when the load instruction references the location using the other register (Wing et al., column 22, lines 29-45, column 23, lines 2-45). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the invention of Katzman, utilize the translation lookaside buffer, as taught by Wing et al., as it's an easy way to keep track of stack references.

Response to Arguments

48. Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

49. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure as follows. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

a. Moore et al., US Patent 5,659,703, have taught a microprocessor system with hierarchical stack and method of operation..

50. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tonia L. Meonske whose telephone number is (571) 272-4170. The examiner can normally be reached on Monday-Friday, with every other Friday off.

51. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie P. Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

52. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

tlm

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100